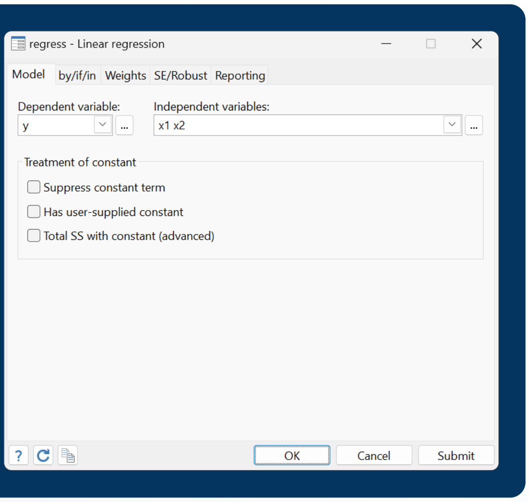
STATA Features

Maximum likelihood estimation

- Over a thousand built-in estimators
- Intuitive and consistent syntax
- Survey data support
- Program your own estimators
- Numerical or analytical derivatives
- Standard errors: OIM, OPG, Robust–Huber/White/sandwich, cluster–robust, bootstrap, jackknife, and more
- Powerful postestimation features
- Use point and click or type commands



regress - Linear regression

Dependent variable: y

Independent variables: x1 x2

Treatment of constant

□ Suppress constant term

□ Has user-supplied constant

□ Total SS with constant (advanced)

Stata offers over a thousand built-in ML estimators

All follow elegant and intuitive syntax and have consistent output. Learn one command, know how to use them all.

Linear regression

```
. regress y x1 x2
```

Logistic regression

```
. logistic y x1 x2
```

Poisson regression

```
. poisson y x1 x2
```

Poisson regression with identity link (GLM)

```
. glm y x1 x2, family(poisson) link(identity)
```

ARIMA/ARMAX

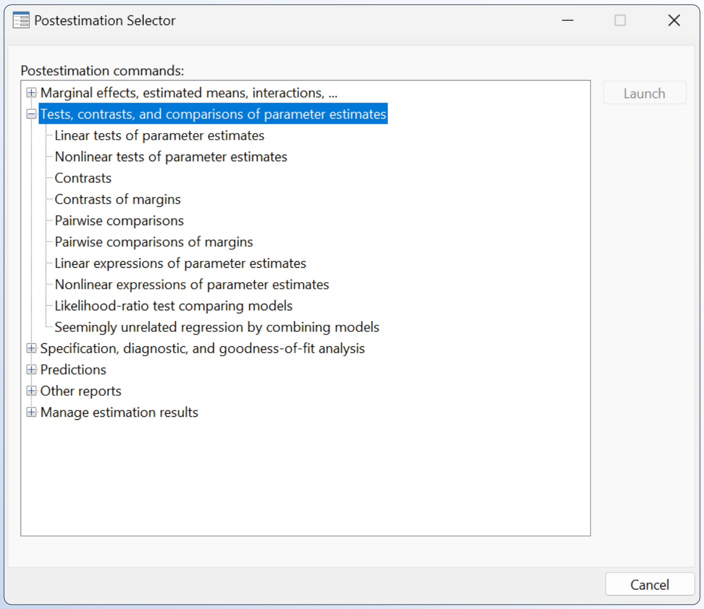
```
. arima y x1 x2, arima(2,1,3)
```

Logistic regression with survey data

```
. svy: logistic y x1 x2
```

After estimation, easily access powerful postestimation features.

```
. postest
```



Postestimation Selector

Postestimation commands:

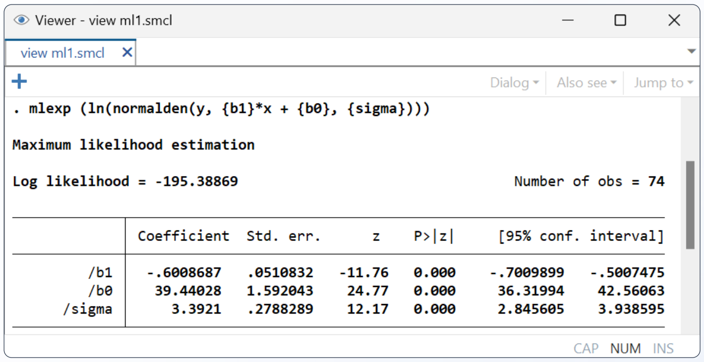
- Marginal effects, estimated means, interactions, ...
- Tests, contrasts, and comparisons of parameter estimates
  - Linear tests of parameter estimates
  - Nonlinear tests of parameter estimates
  - Contrasts
  - Contrasts of margins
  - Pairwise comparisons
  - Pairwise comparisons of margins
  - Linear expressions of parameter estimates
  - Nonlinear expressions of parameter estimates
  - Likelihood-ratio test comparing models
  - Seemingly unrelated regression by combining models
- Specification, diagnostic, and goodness-of-fit analysis
- Predictions
- Other reports
- Manage estimation results

Write your own ML estimators

Stata offers a powerful environment for you to add your own ML estimators. For log likelihoods that can be written as simple expressions, just type the expression in the **mlexp** command. For more complicated expressions, you can write a program in Stata's scripting or matrix language and use the **ml** suite to do the rest for you. You can even turn your ML evaluator into a command.

Type a simple expression

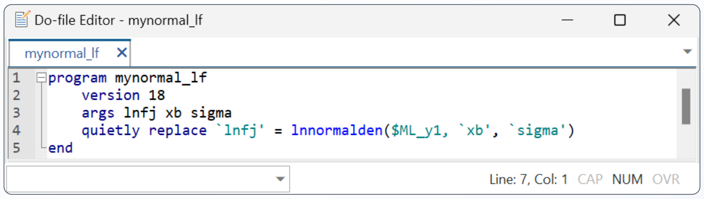
Use **mlexp** when your log likelihood can be expressed simply. For example, for normal linear regression, type



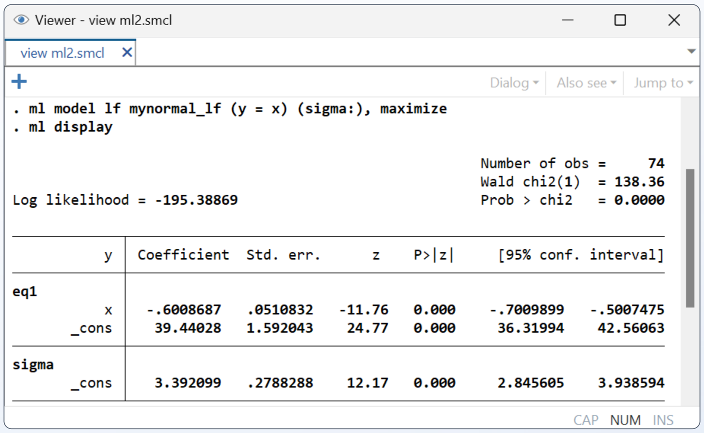
```
. mlexp (ln(normalden(y, {b1}*x + {b0}, {sigma})))

Maximum likelihood estimation

Log likelihood = -195.38869                          Number of obs = 74

------------------------------------------------------------------------------
             |  Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         /b1 |  -.6008687   .0510832   -11.76   0.000    -.7009899   -.5007475
         /b0 |   39.44028   1.592043    24.77   0.000     36.31994    42.56063
      /sigma |     3.3921   .2788289    12.17   0.000     2.845605    3.938595
------------------------------------------------------------------------------
```

Write a program

Write a program to evaluate more complicated likelihood functions.



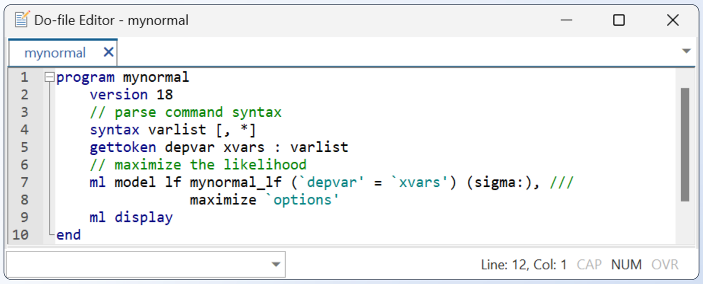
```
program mynormal_lf
    version 18
    args lnfj xb sigma
    quietly replace `lnfj' = lnnormalden($ML_y1, `xb', `sigma')
end
```



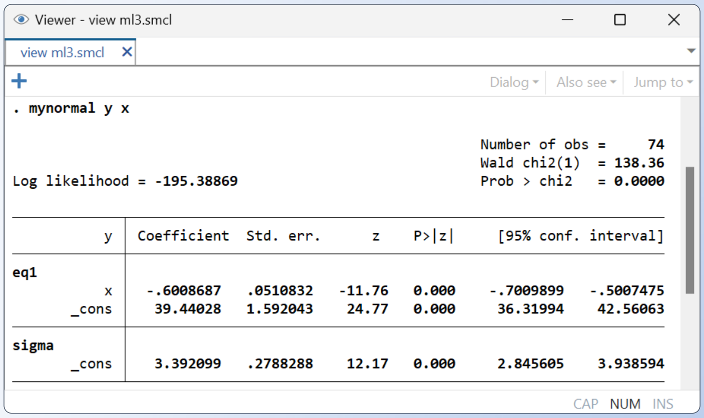
```
. ml model lf mynormal_lf (y = x) (sigma:), maximize
. ml display

                                                       Number of obs =     74
                                                       Wald chi2(1)  = 138.36
Log likelihood = -195.38869                            Prob > chi2   = 0.0000

------------------------------------------------------------------------------
           y |  Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
eq1          |
           x |  -.6008687   .0510832   -11.76   0.000    -.7009899   -.5007475
       _cons |   39.44028   1.592043    24.77   0.000     36.31994    42.56063
-------------+----------------------------------------------------------------
sigma        |
       _cons |   3.392099   .2788288    12.17   0.000     2.845605    3.938594
------------------------------------------------------------------------------
```

Type a simple expression

With another small program, you can turn your likelihood-evaluation program into a full-fledged Stata command.



```
program mynormal
    version 18
    // parse command syntax
    syntax varlist [, *]
    gettoken depvar xvars : varlist
    // maximize the likelihood
    ml model lf mynormal_lf (`depvar' = `xvars') (sigma:), ///
                maximize `options'
    ml display
end
```



```
. mynormal y x

                                                       Number of obs =     74
                                                       Wald chi2(1)  = 138.36
Log likelihood = -195.38869                            Prob > chi2   = 0.0000

------------------------------------------------------------------------------
           y |  Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
eq1          |
           x |  -.6008687   .0510832   -11.76   0.000    -.7009899   -.5007475
       _cons |   39.44028   1.592043    24.77   0.000     36.31994    42.56063
-------------+----------------------------------------------------------------
sigma        |
       _cons |   3.392099   .2788288    12.17   0.000     2.845605    3.938594
------------------------------------------------------------------------------
```

Your new command automatically has many nice features such as options for robust and cluster–robust standard errors without any extra programming effort.

```
. mynormal y x, vce(robust)
. mynormal y x, vce(cluster id)
```

With a few more lines of code, your command can even support survey data,

```
. svy: mynormal y x
```

Your command will also automatically work with postestimation features such as Wald tests, likelihood-ratio tests, contrasts, and much more.

stata.com/mle